Compiler Support for Message Passing Systems

Darpa Contract No. MDA972-97-C-0003

Quarterly R&D Status Report
March 29, 1997 – June 27, 1997

Scientific Computing Associates, Inc.
New Haven, CT

Jens Nielsen, Principal Investigator

July 30, 1997

1. Progress During Reporting Period

In this period, we continued to develop our prototype enhanced MPI tools. We also began to consider better integration of our approach designed with standard MPI in order to make it more attractive to end users. The effort in this period conforms to the approach laid out in our proposal for the core project. The work in this quarter primarily addresses objectives *e.4.1*, *e.4.2*, and *e.4.5* in our work plan.

In addition to the technical work, we attended a number of meetings in relation to the work in this project:

1. In April, Dr. Andrew Sherman, Dr. Robert Bjornson, and Dr. Nicholas Carriero presented a briefing at Darpa headquarters to Dr. Jose Munoz and two of his colleagues on the status of our MPI work.
2. In May, Dr. Sherman and Dr. Sachit Malhotra met with Darpa contractors at Lincoln Laboratory in Lexington, Massachusetts to discuss possible work involving the use of MPI and the enhanced MPI tools developed in this project for some applications in real-time STAP processing. We anticipate that a suitable arrangement will be negotiated to move forward with work in this area. In addition, at the suggestion of David Martinez at Lincoln Laboratory, we have submitted a presentation for the High-Performance Embedded Computing Workshop to be held at Lincoln Laboratory in September.
3. In June, Dr. Daya Atapattu represented Scientific Computing Associates, Inc. at a meeting of the MPI/RealTime Forum in Lexington, Massachusetts. He attended the meeting to help us assess the fit of our work with the proposed MPI/RT standard.

1.1 Enhanced MPI Prototype Development

We continued to develop our MPI prototype software, implementing a number of changes in the system. The most important ones involved MPI derived types, communication safeguards and "shadow communicators."

1.1.1 MPI Derived Types

We developed a new implementation of the transport layer for our system based on the concept of MPI derived types. In our original implementation, an enhanced MPI operation such as `_Send(parm1, vec1:len, parm2)` is implemented using `MPI_Pack()` and `MPI_Unpack()` operations to gather the payload data (two scalars and an array of length `len` in this case) before transmission and to scatter it on receipt. This entails explicit copy operations which may be expensive for large payloads. We have developed an alternative implementation based on dynamic creation of MPI derived datatypes corresponding to the exact data patterns in the message payloads. Instead of using explicit copies to pack and unpack the payloads, we simply compute and send the derived type information and then implement the actual transmissions using the derived types.

In many MPI implementations, the new approach can avoid some of the cost associated with local copy operations. The penalty is that there are now extra messages (to send the derived type information), but we expect to develop optimizations to reduce this overhead. In any event, for large payloads, the benefits from avoiding the extra copy operations far outweigh the costs of the extra communication. In order to provide some optimization, we have implemented an adaptive approach that uses the pack/unpack protocol for short payloads and the derived-types protocol for large ones.

1.1.2 Communication Safeguards

We have implemented a number of safeguards for enhanced MPI communications. For example, we now check datatypes so that we can be sure that matching `_Send()` and `_Recv()` operations both view the data in the same way. We have also implemented size checks to make sure that sufficient buffer space is available in the receiving process to hold the message. These features have been implemented using extra ("out of band") descriptive messages and/or by adding header information to the message payloads, depending on the situation.

1.1.3 Shadow Communicators

In order to implement the two features just described while maintaining the user's ability to mix enhanced MPI communications with ordinary MPI communication (which we ignore at present), we have introduced the concept of "shadow communicators." As noted above, both new features may require that additional messages be transmitted from the sender to the receiver. These extra messages cannot use the original communicator specified by the user, since they are invisible to the user and would interfere with the expected message sequence on the communicator.[1] For each user communicator, therefore, we now create a shadow communicator that we use for all

[1] The effect would be that one of the user's standard `MPI_Recv()` operations might receive one of the "invisible" descriptive messages. This could have several nasty side effects: The unexpected message would confuse the user's code, and our runtime system would probably fail because it had failed to receive the necessary descriptive information.

the messages we need for runtime implementation of our enhanced send and receive operations. One side effect in our prototype (which we may correct in the future) is that data sent by an enhanced send operation must be received by an enhanced receive operation.

In order to implement the shadow communicator approach, we will eventually intercept all calls to the MPI communicator construction functions (`MPI_Comm_create`, `MPI_Comm_split`, `MPI_Comm_dup`, `MPI_Comm_free`, and `MPI_Init`) so that we can create the shadow communicators dynamically. For now, since we wanted to test out the implementation quickly, we have provided alternative functions for each of the construction functions. These functions (`LMPI_Comm_create`, `LMPI_Comm_split`, `LMPI_Comm_dup`, `LMPI_Comm_free`, and `LMPI_Init`, respectively) use the standard MPI functions to create both communicators and set up data structures so that the runtime system knows about the shadow communicators.

1.2 Integration with Standard MPI

In discussing our work with a number of potential users and vendors of MPI technology, it has become increasingly clear that there would be significant added value if we were able to directly parse and process standard MPI in addition to dealing with the "higher level" `_Send()` and `_Recv()` operations. This would make it possible for users to get the benefits of our approach on pre-existing MPI programs without making any modifications to them. Eventually, this would enhance our ability to attack the commercial marketplace by reducing a barrier to product acceptance. During this period, we began to study the issues entailed in dealing with standard MPI operations. We anticipate that there may be significant difficulties related to parser construction, particularly in C, where functions may called indirectly using function pointers.

2. Planned Activities and Milestones

We will continue the development of our MPI prototype software, focusing particularly on improvements in the runtime implementation and on better integration with standard MPI. We continue to discuss possible "beta" use by a number of users, including Lincoln Laboratory and Concurrent Technologies Corporation, both of whom are Darpa contractors.

As noted, we have also begun an interaction with the MPI/Real-Time Forum. While our initial efforts are at a low level, we intend to continue tracking the evolving MPI/RT standard and plan to get more actively involved as appropriate, if funding can be arranged when required. In this regard, we hope to begin some contract work with Lincoln Laboratory, and we expect to attend the HPEC Meeting in Massachusetts in September.

We continue to be interested in arranging for early exercise of the two options associated with the project, since we believe that Darpa would see significant benefits if our work were to be accelerated. From discussions with Dr. Jose Munoz, the Darpa program manager for this project, we understand that financial exigencies at Darpa make such exercise impossible at the present time, but we hope to renew consideration for exercise at the start of FY1998.

3. Administrative Information

No significant problems have arisen in this period, and there are no areas of concern. The core portion of the project is ahead of schedule with respect to technical development, and the cost is consistent with the expenditure plan. There were no changes in key personnel during this period, and there were no purchases of major equipment in this period.

Personnel Hours		
	Planned	Actual
Current Period	762	762.5
Contract Since Inception	1595	1595.5

Expenditures in current period: $ 86,535 (inclusive of fee)

Expenditures since inception: $ 176,290 (inclusive of fee)

Total funds committed: $ 374,733

Estimated funds for completion: $ 198,443

Approximate quarterly breakout of anticipated payments from DARPA:

$ 45,000 per calendar quarter through 2Q1998;
$ 60,000 in 3Q1998;
$ 16,211 in 4Q1998.

Estimated date of completion: October 15, 1998